

伍、中文發明摘要：

本發明揭露一種以直接螢幕顯示(OSD)方法在一影音解碼/播放/複製系統的顯示裝置上顯示雙位元字集(double byte character set, DBCS)的方法，以及利用此種方法來更新顯示畫面的影音解碼/播放/複製系統，其係在該儲存媒體中儲存至少一種雙位元字集檔案以及影音來源檔案，讀取該儲存媒體之檔案系統並取得影音來源檔案的檔案名稱的長度，由該檔案系統取得該儲存媒體各個雙位元字集檔案之內碼，並計算所需顯示字元的內碼於該儲存媒體中之實體位置，以依據該實體位置自相對應的字集檔案中提取相對應該內碼的字元，最後以該字元更新原有的 OSD 資料並更新一顯示器的顯示畫面。

陸、英文發明摘要：

柒、指定代表圖：

(一)本案指定代表圖為：第（ 5 ）圖。

(二)本代表圖之元件代表符號簡單說明：

501 步驟

504 步驟

502 步驟

505 步驟

503 步驟

506 步驟

捌、本案若有化學式時，請揭示最能顯示發明特徵的化學式：

玖、發明說明：

【發明所屬之技術領域】

本發明是有關於一種具備能顯示雙位元字集(double byte character set, DBCS)之直接螢幕顯示(on screen display, OSD)功能之影音解碼/播放/複製系統及利用直接螢幕顯示技術顯示雙位元字集以更新影音解碼/播放/複製系統的顯示畫面的之方法，特別是指一種不需要增加光學解碼系統中之記憶體大小，即能顯示雙位元字集之光學系統及其方法。

【先前技術】

直接螢幕顯示(以下簡稱 OSD)在消費性電子產品中佔據相當重要的地位，一般常用來提供互動資訊，例如系統設定、導引目錄等，供使用者直接進行系統組態的設定與控制。常見之 OSD 應用例是在個人電腦、電視機上，用來提供畫面資訊，並能供使用者對畫面顯示的相關參數進行直接調整。

如圖 1 所示，為一視訊處理系統 500，主要包含有一處理器 10、一 OSD 單元 20、一視訊解碼器 30、一混合器(mixer)40、一唯讀記憶體 50、一動態隨機存取記憶體 60(以下簡稱 DRAM)，以及一 OSD 緩衝記憶體 70。

處理器 10 可以讀取一儲存媒體(圖未示)中所儲存的資料，例如經壓縮後之視訊資料，當處理器 10 接收到視訊資料之位元流(bistreams)後，即會傳送至視訊解碼器 30 上，由視訊解碼器 30 進行解碼的動作，解碼後的畫面(frame)會暫儲在 DRAM 60 中，以便於重建出原始的畫面後傳送至顯

示器 80 上供使用者觀看。

唯讀記憶體 50 中儲存有複數電腦程式碼，是用來進行此視訊處理系統 500 開機時的初始化，以及視訊解碼等邏輯演算。OSD 緩衝記憶體 70 中則儲存有包含 OSD 檔頭 (OSD header)、OSD bitmap 以及圖示等等所構成的 OSD 資料，其中已指定了 OSD 資料出現在顯示器 80 上的座標位置、明亮度，以及每一畫素的色彩索引值(indcx)等等。

當使用者啟動 OSD 功能時，是以一 OSD 啟動鍵(圖未示)來控制處理器 10，進而使 OSD 單元 20 去存取 OSD 資訊，以建構出 OSD 畫面，並當 OSD 畫面被傳送至混合器 40 上後，藉由混合器 40 來疊加 OSD 畫面與經視訊解碼器 30 解碼後之原始影像畫面，就能夠產生一包含有 OSD 資訊的畫面。

一般顯示在顯示器 80 畫面上的 OSD 資訊，除了用圖示來表達其意義以外，文字的輔助說明也甚為重要，故會在唯讀記憶體 50 中儲存相關字型來供 OSD 單元 20 透過處理器 10 進行讀取，以作為顯示時文字的來源依據。由於唯讀記憶體 50 之造價不菲，故為降低整體成本起見，皆僅儲存特定的文字或字元(26 個英文字母，以及些許特定文字、符號…)，以避免需要採用高容量的唯讀記憶體 50。

上述之視訊解碼系統 500，其 OSD 的另一種應用是以唯讀記憶體 50 中的程式碼來讀出儲存媒體中的目錄名稱與檔案名稱，以便使用者毋需透過個人電腦，僅需利用 OSD 就能觀看到儲存媒體中所儲存的檔案，然而因為唯讀記憶

體 50 中的字型有限，故往往僅能見到特定字元所建置的檔案名稱，若是以非拼音語系之內碼系統，例如中文、韓文、日文…等等需要超過一個的位元組來表示字元的雙位元字集語言系統所構成之檔案名稱，則因唯讀記憶體 50 中並未儲存有該內碼系統中所定義之全部字元，往往造成檔名中的某些字元無法正確顯示。

【發明內容】

有鑑於以往的缺點，本發明之目的即在提供一種毋需額外的記憶體空間，即能以直接螢幕顯示(OSD)方式來顯示雙位元字集方法以及影音解碼/播放/複製系統。

於是，在一較佳實施例中，本發明之影音解碼/播放/複製系統具有一直接螢幕顯示控制裝置，此控制裝置能讀取一儲存媒體中的雙位元字集檔案，並以直接螢幕顯示方式將雙位元字集顯示該檔案結構於一顯示器上以更新顯示器的顯示畫面，其中該控制裝置包含一 OSD 緩衝記憶體、一記憶體，以及一 OSD 單元。

該 OSD 緩衝記憶體中儲存有 OSD 資料，該影音解碼/播放/複製系統的處理器是與該記憶體相耦接，該處理器能讀取該儲存媒體之該檔案系統，並且計算儲存媒體中的影音來源檔案之名稱的長度。該 OSD 單元是與該 OSD 緩衝記憶體相耦接，該 OSD 單元能經由該處理器，依據該內碼自該字型檔中擷取出與該內碼相對應之字元後，配合該 OSD 資料，以直接螢幕顯示方式將該檔案名稱顯示於該顯示器上並更新該顯示器的顯示畫面。

在該較佳實施例中，依據上述架構，本發明亦提供一種利用以直接螢幕顯示方式來顯示出該檔案系統之方法，其包含步驟：讀取該儲存媒體之一檔案系統並計算該儲存媒體的影音來源檔案的檔案名稱的長度，接著由該檔案系統取得構成該檔案之名稱的內碼，並計算各該內碼於該儲存媒體中的雙位元字集檔案之實體位置後，由各該實體位置擷取與各該內碼相對應的字元，最後依據該等內碼自該雙位元字集檔案檔中取得相對應之字元，使得構成該名稱之該等字元顯示在一顯示器上。

本發明的影音解碼/播放/複製系統將雙位元字集檔案儲存於儲存媒體中，不需要額外的記憶體空間就可以儲存有各式的字型，因此不論以何種內碼系統儲存的檔案系統，都能夠藉由正確地計算出每一字元的實體位置後，自儲存媒體中取得每一內碼所相對應的字元後顯示在顯示器上，因此不論以何種內碼系統所建置之檔名都能夠被正確地顯示。

【實施方式】

有關本發明之前述及其他技術內容、特點與功效，在以下配合參考圖式之一較佳實施例的詳細說明中，將可清楚的明白。

參閱圖 2，本發明影音解碼/播放/複製系統 100 是能將一儲存媒體之檔案結構，例如目錄名稱、檔案名稱等，以直接螢幕顯示(以下簡稱 OSD)方式顯示在一顯示器 300 上。

本實施例中，儲存媒體是一光碟片 200，例如 CD-R、

CD-RW、DVD、DVD-R、DVD+R、DVD RAM、DVD- RW
、DVD-RW…等各式之光學儲存媒體。此儲存媒體中並儲
存有依據 MPEG 標準規範所壓縮的影像或聲音檔案，例如
副檔名為 mpg 之影音檔案，或如副檔名為 mp3 之聲音檔案
5，但並不僅以此為限，例如為一般影音播放裝置所支援的
其它影音檔 (*.wmv,*.wma,*.wav,*.jpg,*.mid,...)。需說明
的是，本實施例中之檔案是以繁體中文來作為其檔名的編
碼方式，因此其檔名無法在以往的解碼系統中透過 OSD 方
式來顯示。另外，運用於光學儲存媒體上之 MPEG 標準通
常可分為 MPEG-1(ISO/IEC 11172)，以及 MPEG-2(ISO/IEC
10 13818)兩種，前者是用在一般 VCD 光碟片上，後者則是用
在 DVD 或 SVCD 光碟片上。此外，儲存媒體亦可由記憶卡
700 來實現，兩種儲存裝置皆可用來實施本發明的技術。

此外，儲存媒體(200,700)中並儲存有複數種雙位元字
集(DBCS)字型檔，每個字型檔是對應著不同的內碼系統，
15 並具有該內碼系統所指定之所有字元，本實施例中，僅以
五種字型檔來作說明，但不應以此作為本實施例之限制。
五種字型檔之檔名分別為：big5.24、gbk.24、unicode.24，
ksc5601.24，以及 shift-jis.24，其中，big5.24 是對應 BIG5
20 之內碼系統，約有一萬三千個繁體中文字元，gbk.24 是對
應 GBK 之內碼系統(簡體中文)，unicode.24 是對應於
Unicode 2.0 之內碼系統(11172 個韓文字元)，ksc5601.24
是對應於 ksc5601 之內碼系統(11172 個韓文字元)，而 shift-
jis.24 是對應於 Shift-JIS 日文字型的編碼系統。

參閱圖 3，顯示的是 big5.24 字型檔中，BIG5 編碼系統下所有繁體中文字元的邏輯對應表，如圖所示，BIG 編碼系統是以起始位置為「A140」(以 16 進制表示)至終止位置為「F9FE」間的位置編排每一繁體中文字元的內碼，然而，其間並非所有的位置都有相對應的中文字元，可能為特殊符號，或可能是供使用者自行定義的空白字元，其中，第一個中文字元「一」的內碼位置是在「A440」，第二個中文字元「乙」的內碼位置是在「A441」，第三個中文字元「丁」的內碼位置是在「A442」，其餘類推。

在檔案的實體存放內容中，每一個中文的內碼(2 個位元組)後方緊接著是一個以 72 位元組來表示的繁體中文字元(72 位元組=24 bits×24 bits，此字元顯示在顯示器 300 上的大小為 24×24)，因此必需得知其內碼後，才能由該內碼來取得相對應的字元。

若儲存媒體(200,700)係選擇為光碟片 200，其使用之檔案系統是 Juliet，因此可以支援檔名為繁體中文的檔案。如吾人所熟知，其它諸如 Romeo、UDF，以及 ISO 9660 等，皆為光學儲存媒體所能應用之檔案系統，其中，ISO 9660 檔案系統特別可適用於各式的作業系統，然其缺點為僅支援 8 位元之檔名，以及 3 位元的副檔名。藉由檔案系統的讀取，就可以得知儲存媒體中每個檔案的實體存放位置、路徑、目錄，以及名稱等資訊。此外，上述各字型檔之副檔名僅是用來與其它種類的檔案來作明顯區隔，其副檔名的指定可依設計時的不同而加以改變，並不僅限於上述之

副檔名「24」。

5 影音解碼/播放/複製系統 100 主要包含了一光學讀寫頭
1、一處理器 2、一唯讀記憶體 3、一動態隨機存取記憶體
4(以下簡稱 DRAM)、一控制器 5、一視訊處理器 6、一音訊
處理器 7、以及一 OSD 緩衝記憶體 8 等等。本發明尚包含
一影音來源及 OSD 字型檔偵測模組 9、其係用來偵測影音
來源檔案及 OSD 字型檔(雙位元字集檔案)的位置，並在當
有數種型式的儲存媒體記錄影音來源及 OSD 字型檔可以獲
得時，切換影音來源檔案及 OSD 字型檔的位置。若是採用
10 光碟片 200 為記錄影音來源檔案與雙位元字集檔案的儲存
媒體(200,700)，則需透過光學讀寫頭 1 讓處理器對其進行
資料的讀取與寫入。處理器 2 是數位解碼系統 100 中之控
制核心，負責協調各構件間的通訊與資料傳輸，亦能受使
用者控制而執行特定功能，例如開、關機、啟動 OSD 功能
15 ...。

20 唯讀記憶體 3 中儲存有複數電腦程式碼，供處理器 2
讀取至 DRAM 中，以便於能夠快速地執行特定功能，依其
功能劃分，其具有一檔案過濾模組 31 以及一字元計算模組
32，檔案過濾模組 31 之功能乃在讀取儲存媒體(200,700)之
檔案系統，並辨別出不同檔案之副檔名，以便擷取出所需
之檔案，例如雙位元字集檔案。字元計算模組 32 則能自雙
位元字集檔案中，計算出所需之字元在儲存媒體中的實體
位置，並依此位置取得所需字元，此部分將於下文中再行
詳述。

控制器 5 是受處理器 2 之控制，並能接收來自於光學讀寫頭 1 自儲存媒體 200 中讀取出的資料，例如包含了影像與聲音之 MPEG-2 位元流(bitstream)，控制器 5 之功能類似於一解多工器(demux)，能夠將影像資料與聲音資料分離，並分別交由視訊處理器 6 與音訊處理器 7 加以處理後輸出。

視訊處理器 6 主要在進行 MPEG-2 的解碼工作，依序進行可變長度解碼(VLD)、反量化運算(inverse Quantization)、逆向離散餘弦轉換編碼(IDCT)、動態補償(motion compensation)等動作，就能夠將影像位元流轉換為原始的影像資料，最後再經編碼為 NTSC 或 PAL 格式之畫面，就能夠輸出至顯示器 300 上。

音訊處理器 7 也是進行與視訊處理器 6 相類似的工作，當聲音信號被解壓縮還原後，再經過數位/類比轉換，就能輸出至揚聲器 400 上，進行聲音的輸出。

控制器 5 具有一 OSD 單元 51，其利用 OSD 緩衝記憶體 8 來儲存 OSD 資料與相關的 OSD 規格，參閱圖 4，OSD 資料主要包含有檔頭(header)81、色盤(palette)82，以及畫素資料(pixel data)83 等，其中，檔頭 81 記錄了 OSD 顯示的區域座標、功能碼(function code)以及色盤的資料輸入(entry)，功能碼指定 OSD 是採取 4 色、16 色或 256 色哪一種模式顯示，色盤 82 記錄了每一個 OSD 畫素的亮度與彩度，畫素資料 83 則記錄每一畫素的色彩索引值，以便於正確地顯示在顯示器 300 的特定區域。

當使用者啟動 OSD 功能後，OSD 單元 51 即回應處理器 2 之要求，會自 OSD 緩衝記憶體 8 中讀出 OSD 資料，並依據使用者所指定之 OSD 選項(例如顯示選單)來處理特定的 OSD 資料，隨後 OSD 單元 51 即把處理完畢之 OSD 資料傳送至視訊處理器 6 上，使其與解碼後之畫面疊合，並傳送顯示器 300 上供使用者觀看。

參閱圖 5，顯示的是本發明以直接螢幕顯示的方法顯示雙位元字集以更新顯示畫面的程序。如步驟 501，在儲存媒體(200,700)中儲存雙位元字集檔案及影音來源檔案，此字型檔需與此儲存媒體(200,700)之檔案系統相配合，例如儲存媒體(200,700)是以繁體中文來記錄目錄、檔案的名稱，則至少有一字型檔是用來儲存繁體中文之字元，本實施例中是儲存有五種字型檔：big5.24、gbk.24、unicode.24、ksc5601.24 以及 shift-jis.24 來支援繁體中文、簡體中文、韓文以及日文等三種語系。至於將字型檔儲存於儲存媒體(200,700)中的時機，可以是空白的儲存媒體出廠時內建於其中，也可以是使用者利用電腦(圖未示)將檔案燒錄於儲存媒體(200,700)時一併寫入，由於字型檔之大小一般僅在 400~1MB 之間，故僅佔儲存媒體空間的極小部分。

在步驟 502 與 503 中，當使用者啟動 OSD 後，處理器 2 即會讀取儲存媒體(200,700)之檔案系統，因此 OSD 單元 51 可以得知儲存於儲存媒體(200,700)中之目錄名稱及檔案名稱並且計算儲存媒體中的影音來源檔案的檔案名稱的長度。

在步驟 504 中，由於 OSD 緩衝記憶體 8 或唯讀記憶體 3 中並未儲存有繁體中文字元，因此 OSD 單元 51 必需依據前述檔案名稱之內碼，由儲存媒體(200,700)之字型檔中提取相對應之字元。因為處理器 2 已藉由檔案過濾模組 31 讀取出所有副檔名為「24」之檔案，故知至少能支援五種不同的內碼系統：BIG-5(繁體中文)、GBK(簡體中文)、Unicode 2.0(韓文)、ksc5601(韓文)以及 Shift-JIS(日文)。

經由處理器 2 自行偵測，或是由使用者加以指定，就能知道目前是要由哪一個雙位元字集檔案中提取相對應的字元。在進行相對應的字元讀取前，雖已知所欲字元之內碼，但由於雙位元字集檔案記錄字元的方式是在每一內碼後再記錄其相對應字元的資料(72 位元組)，即每一內碼的實體位置會較前一內碼偏移有 72 個位元組，故在知道此所需字元的內碼後，尚需計算出所需字元之總偏移量，才能由字型檔的實體位置加上總偏移量，以得到該字元的實體位置。

由於每一種內碼系統所定義之內碼不同，因此其計算總偏移量之程式亦不相同，在此分別列舉本實施例中各字型檔計算總偏移量之程式及設計說明係列示如下：

(1)繁體中文(BIG5)：

```
Set Code          /*設定內碼值 */
Set Code1, Code2   /*設定內碼值之高、低位元組 */
Set i, j           /* 設定暫存變數 */
Set Total-Offset   /*設定總偏移量 */
```

```

If (Code  $\geq$  0xA140 && Code  $\leq$  0xF9FE) /* 判斷內碼值是否在
    0xA140 與 0xF9FE 的區間中 */
{
code1=(Code & 0xFF00)>>8; /* 取內碼值的高位元組 */
5 code2=(Code & 0xFF);      /* 取內碼值的低位元組 */
i=(code1-0xA1);             /* 計算內碼值的高位元組與初始內碼
    值高位元組間的差距 */
j=(code2-0x40);             /* 計算內碼值的低位元組與初始
    內碼低位元組間的差距 */
10 if(code2 > 0xA0)
j=j-34; /* 由於內碼值的低位元組在 0x7F 至 0xA0 間共有 34 個
    位置並未編碼，故需將計算出來的內碼值的低位元組與初始
    內碼低位元組間的差距減去 34 以便將空白編碼位置忽略不計
    */
15 Total-Offset=2+(i  $\times$  (12  $\times$  16-35)  $\times$  74)+(j  $\times$  72)+(j  $\times$  2) ;
    ))))))))))))))))))))))))))))))))))))))))))))))))))))))))
    ))
/* 根據內碼值的高位元組與初始內碼值的高位元組間的差距，
    內碼值的低位元組與初始內碼值的低位元組間的差距減去空
20 白編碼位置數量的數值，以及字集位元組數目(2 bytes)加上
    字型資料位元組數目(72 bytes)的總和計來算總偏移量 */
舉例來說，Big5 繁體中文編碼系統乃是從 0xA140 至 0xF9FF 開
    始編碼，因此其初始內碼值係設定為 0xA140 而終止內碼值
    為 0xF9F。該方法的執行步驟一開始先設定所有運算參數以

```

及暫存變數，並且判斷想要的雙位元字元的內碼值是否落於 Big5 繁體中文編碼系統的編碼範圍內。若是，則分別擷取出內碼值的高位元組與低位元組，並且分別計算內碼值的高位元組與初始內碼值的高位元組間的差距，以及計算內碼值的低位元組與初始內碼值的低位元組間的差距。接下來，由於由於內碼值的低位元組在 0x7F 至 0xA0 間共有 34 個位置並未編碼(null character)，故需跳過不計並將計算出來的內碼值的低位元組與初始內碼值的低位元組間的差距再減掉 34。最後，根據想要的雙位元字元的內碼值的高位元組與初始內碼值的高位元組間的差距，想要的雙位元字元的內碼的內碼值的低位元組與初始內碼值的低位元組間的差距減去空白編碼位置數量的數值，以及雙位元字元位元組數目加上字型資料位元組數目的總和計來算想要的雙位元字元於該儲存媒體中之實體位置。以「一」字元為例，其高位元組即是指四個內碼中之前兩個內碼「A4」，低位元組是指後兩個內碼「40」，其計算方式即是由 BIG5 之內碼系統所定義的起始位置「A140」、終止位置「F9FE」，計算出其總偏移量，即能得知每一字元儲放的實體位置，下文中其餘數種內碼系統之計算方式亦大致相同。

20 (2)簡體中文(GBK)：

Set Code

Set Code1，Code2

Set i，j

Set Total-Offset

```

If (Code  $\geq$  0x8140 && Code  $\leq$  0xFEFF) /* 判段內碼值是否在
    0x8140 與 0xFEFF 的區間中 */
{
code1=(Code & 0xFF00)>>8; /* 取內碼值的高位元組 */
5 code2=(Code & 0xFF); /* 取內碼值的低位元組 */
i=(code1-0x81); /* 計算內碼值的高位元組與初始
    內碼值高位元組間的差距 */
j=(code2-0x40); /* 計算內碼值的低位元組與初始
    內碼低位元組間的差距 */
10 if(code2 > 0x7F)
j=j-1; /* 由於內碼值的低位元組在 0x7F
    這個位置並未編碼，故需將計算出來的內碼值的低位元組與
    初始內碼低位元組間的差距減去 1 以便將這個空白編碼位置
    忽略不計 */
15 Total-Offset=2+(i×(12×(16-2)×74)+(j×72)+(j×2); )))))))))).
/* 根據內碼值的高位元組與初始內碼值的高位元組間的差距，
    內碼值的低位元組與初始內碼值的低位元組間的差距減去空
    白編碼位置數量的數值，以及字集位元組數目(2 bytes)加上
    字型資料位元組數目(72 bytes)的總和計來算總偏移量 */
20 (3)韓文(Unicode 2.0):
Set Code
Set Code1, Code2
Set i, j
Set Total-Offset

```

```

If (Code  $\geq$  0xAC00 && Code  $\leq$  0xD7A3) /* 判段內碼值是否在
    0xAC00 與 0xD7A3 的區間中 */
{
    code1=(Code & 0xFF00)>>8;    /* 取內碼值的高位元組 */
5   code2=(Code & 0xFF);          /* 取內碼值的低位元組 */
    i=(code1-0xAC);               /* 計算內碼值的高位元組與初
        始內碼值高位元組間的差距 */
    j=(code2-0x00);               /* 計算內碼值的低位元組與初
        始內碼低位元組間的差距 */
10  Total-Offset=2+(i*(16*16)*74)+(j*72)+(j*2); }。
    /* 根據內碼值的高位元組與初始內碼值的高位元組間的差距，
        內碼值的低位元組與初始內碼值的低位元組間的差距減去空
        白編碼位置數量的數值，以及字集位元組數目(2 bytes)加上
        字型資料位元組數目(72 bytes)的總和計來算總偏移量 */
15  (4)韓文(Johab/KSC05601-1992):
    Set Code
    Set Code1, Code2
    Set i, j, k, m, n, p
    Set Total-Offset
20  If (Code  $\geq$  0x8861 && Code  $\leq$  0xD3BD)
    {
        code1=(Code & 0xFF00)>>8;
        code2=(Code & 0xFF);
        i=(code1-0x88);

```



```

m=(i%4) ;
k=(i>>2)
j=(code-0x8860)-(k×1024) ;
n=(j+32)-(2×m) ;
5  p=code2 >>4 ;
q=(code2 & 0xf)
if ((p%2)!=0)
{
if (q>2)
10 Total-Offset=2+[(k×588)+(j-2-(m×64)-(n×4))]×74 ;
else
Total-Offset=2+[(k×588)+(j-1-(m×64)-(n×4))]×74 ;
}
else
15 Total-Offset=2+[(k×588)+(j-1-(m×64)-(n×4))]×74 ; }。

```

(5) 日 文 (Shift-JIS) :

(1) If (Code \geq 0x889F && Code \leq 0x9872) /* 判段內碼值
 是否在 0x889F 與 0x9872 的第一區間中 */

```

20 {
code1=(Code & 0xFF00)>>8 ; /* 取內碼值的高位元組 */
code2=(Code & 0xFF) ; /* 取內碼值的低位元組 */
i=(code1-0x88) ; /* 計算內碼值的高位元組與初始內碼值高

```

位元組間的差距 */

j=(code2-0x40); /* 計算若內碼值在 0x8940 ~ 0x9872 之部

分區間時，內碼值的低位元組與與此區間之初始內碼值的低

位元組間的差距 */if (i == 0) /* 內碼值的範圍在 0x889F ~

5 0x88FC 之間 */

{

k = (code2 - 0x9F); /* 計算內碼值與初始內碼值間

的位移 */

}

10 else /* 內碼值的範圍在 0x8940 ~ 0x9872 之間 */

{

if (code2 > 0x7F)

{

k = 94 + (i - 1) × (10 × 16 + 15 + 13) + (j - 1);

15 /* 計算內碼值與初始內碼值間的位移 */

}

else

{

k = 94 + (i - 1) × (10 × 16 + 15 + 13) + j ;

20 /* 計算內碼值與初始內碼值間的位移 */

```
}
```

```
}
```

```
Total-Offset = 2 + ( k × 74 ); /* 根據字集位元組數目 (2
```

```
bytes)加上字型資料位元組數目 (72 bytes)的總和與內碼值與初
```

```
5 始內碼值之間的位移來計算總偏移量 */
```

```
}
```

```
/* 計算字集範圍從 0x989F ~ 0x9FFC */
```

```
10 (2) If (Code ≥ 0x989F && Code ≤ 0x9FFC) /* 判段內碼值  
是否在 0x989F 與 0x9FFC 的區間中 */
```

```
{
```

```
code1=(Code & 0xFF00)>>8; /* 取內碼值的高位元組 */
```

```
code2=(Code & 0xFF); /* 取內碼值的低位元組 */
```

```
15 i=(code1-0x98); /* 計算內碼值的高位元組與初  
始內碼值高位元組間的差距 */
```

```
j=(code2-0x40); /* 計算若內碼值在 0x9940 ~
```

```
0x9FFC 之區間時，內碼值的低位元組與與此區間之初始內碼  
值的低位元組間的差距 */
```

```
20 if ( i == 0 ) /* 內碼值的範圍在 0x989F ~ 0x98FC 之間 */
```

$k = 2965 + (\text{code2} - 0x9F)$; /* 2965 是從(1)式子得來的：將 0x9872 的內碼值代入(1)式得從 0x889F ~0x9872 位置的偏移量為 2965 */

else /*內碼值的範圍在 0x9940 ~ 0x9FFC 之間 */

5 {

if (code2 > 0x7F)

{

$k = 2965 + 94 + (i - 1) \times (10 \times 16 + 15 + 13) + (j - 1)$; /* 根據字集位元組數目(2 bytes)加上字型資料位元組數目(72 bytes)的總和與內碼值與初始內碼值之間的位移來計算總偏移量 */

}

else

{

15 $k = 2965 + 94 + (i - 1) \times (10 \times 16 + 15 + 13) + j$; } /* 計算內碼值與初始內碼值間的位移 */

}

Total-Offset = $2 + (k \times 74)$; /* 74 表示字集(2 bytes)+字型 data 是 72 (bytes)=74 (bytes) */

20 } /* 根據字集位元組數目(2 bytes)加上字型資料位元組數目

(72 bytes)的總和與內碼值與初始內碼值之間的位移來計算總
偏移量 */

(3)

5 /* 計算字集範圍從 0xE040 ~ 0xEAA2 */

If (Code \geq 0xE040 && Code \leq 0xEAA2) /* 判段內碼值是
否在 0xE040 與 0xEAA2 的區間中 */

{

code1=(Code & 0xFF00)>>8; /* 取內碼值的高位元組 */

10 code2=(Code & 0xFF); /* 取內碼值的低位元組 */

i=(code1-0xE0); /* 計算內碼值的高位元組與初
始內碼值高位元組間的差距 */

j=(code2-0x40); /* 計算內碼值的低位元組與初
始內碼值低位元組間的差距 */

15

if (code2 > 0x7F)

{

k = 4375 + i \times (10 \times 16 + 15 + 13) + (j - 1);

}

/* 4375 是從(2)式子得來的：將內

20 碼值 0x9FFC 代入(2)式得到從 0x889F ~0x9FFC 的總偏移量為

4375 */

else

{

$k = 4375 + i \times (10 \times 16 + 15 + 13) + j;$

5 } /* 計算內碼值與初始內碼值間的位移 */

Total-Offset = 2 + k × 74; /* 根據字集位元組數目(2 bytes)加

上字型資料位元組數目(72 bytes)的總和與內碼值與初始內碼

值之間的位移來計算總偏移量 */

}

10 在步驟 505、506 中，得知所需字元之實體位置後，即自
字型檔中捉取所需之字元資料，而在取得此字元資料後，
OSD 單元 51 即知此字元需在畫面的哪些畫素(24×24bits)需以
特定顏色來顯示為一字元，因此配合 OSD 緩衝記憶體 8 中之
OSD 資料，就能夠更新原有的 OSD 資料並將字元顯示在顯
15 示器 300 上。

綜合上述，本發明將字型檔儲存於儲存媒體中，不需
要額外的記憶體空間就可以儲存有各式的雙字元字集的字
型，因此不論以何種雙字元字集內碼系統儲存的檔案系統
，都能夠藉由正確地計算出每一字元的偏移量，而自儲存
20 媒體中取得每一內碼所相對應的字元，並正確地顯示於顯
示器上，故能以螢幕直接顯示方式(OSD)得知儲存媒體中之
檔案結構，方便使用者進行播放、管理，確實達到本發明

之目的。

惟以上所述者，僅為本發明之較佳實施例而已，當不能以此限定本發明實施之範圍，即大凡依本發明申請專利範圍及發明說明書內容所作之簡單的等效變化與修飾，皆應仍屬本發明專利涵蓋之範圍內。

【圖式簡單說明】

圖 1 是一示意圖，說明以往視訊處理系統中完成直接螢幕(OSD)顯示功能的系統；

圖 2 是一示意圖，說明本發明利用直接螢幕顯示方法來顯示雙位元字集 (DBCS)於之一較佳實施例

圖 3 是一示意圖，說明繁體中文 BIG5 內碼系統之內碼對照表；

圖 4 是一示意圖，說明 OSD 資料結構；以及

圖 5 是一流程圖，說明本發明以直接螢幕顯示(OSD)方法顯示雙位元字集 (DBCS)以更新在一影音解碼/播放/複製系統的顯示畫面的程序。

【圖式之主要元件代表符號說明】

100	數位解碼系統	檔偵測模組
(200,700)	儲存媒體	31 檔案過濾模組
300	顯示器	32 字元計算模組
400	揚聲器	51 OSD單元
1	光學讀寫頭	81 OSD檔頭
2	處理器	82 色盤
3	唯讀記憶體	83 畫素資料
4	動態隨機存取記憶體	501 步驟
5	控制器	502 步驟
6	視訊處理器	503 步驟
7	音訊處理器	504 步驟
8	OSD緩衝記憶體	505 步驟
9	影音來源及 OSD 字型	506 步驟

拾、申請專利範圍：

1. 一種在一影音解碼/播放/複製系統的一顯示裝置上顯示一雙位元字元並藉以更新該顯示裝置的顯示畫面的方法，該方法包含下列步驟：

A)儲存一雙位元字集檔案於以及一影音來源檔案於一儲存媒體中；

B)讀取該儲存媒體的檔案系統的資訊並計算該影音來源檔案之檔案名稱的長度；

C)自該雙位元字集檔案中的取得想要的雙位元字元；以及

D)以取得的雙位元字元更新原始的直接螢幕顯示資料，藉此更新該影音解碼/播放/複製系統的該顯示裝置的顯示畫面。

2. 如申請專利範圍第 1 項所述之方法，其中，該儲存媒體係為光碟片以及記憶卡中其中一種。

3. 如申請專利範圍第 1 項所述之方法，其中，該步驟 C)更包含下列次步驟：

C1)計算想要的雙位元字元於該儲存媒體中之實體位置；以及

C2)由各該實體位置擷取與想要的雙位元字元。

4. 如申請專利範圍第 3 項所述之方法，其中，若該雙位元字集檔案係採用 BIG5 內碼系統或 gbk 內碼系統或 Unicode 韓文內碼系統，該步驟(C1)係以下列步驟來完成：

計算想要的雙位元字元的內碼的高位元組與雙位元字集檔案的初始內碼之間的高位元組差距以及低位元組差距；以及

根據想要的雙位元字元的內碼的高位元組與初始內碼值的高位元組間的差距，想要的雙位元字元的內碼的內碼的低位元組與初始內碼的低位元組間的差距減去空白編碼位置數量的數值，以及雙位元字元位元組數目加上字型資料位元組數目的總和計來算想要的雙位元字元於該儲存媒體中之實體位置。

5. 如申請專利範圍第 3 項所述之方法，其中，若該雙位元字集檔案係採用 Shift-JIS 內碼系統，該步驟(C1)係以下列步驟來完成：

若想要的雙位元字元的內碼落於該內碼系統的第一區間中，計算想要的雙位元字元的內碼的高位元組與該第一區間之初始內碼的高位元組間的差距，計算想要的雙位元字元的內碼的低位元組與該第一區間之初始內碼的低位元組間的差距，並根據想要的雙位元字元的內碼的高位元組與初始內碼值的高位元組間的差距，想要的雙位元字元的內碼的內碼的低位元組與初始內碼的低位元組間的差距減去空白編碼位置數量的數值，以及雙位元字元位元組數目加上字型資料位元組數目的總和計來算想要的雙位元字元於該儲存媒體中之實體位置；

若想要的雙位元字元的內碼落於該內碼系統的第二區間中，計算想要的雙位元字元的內碼的高位元組與該第二區間

之初始內碼的高位元組間的差距，計算想要的雙位元字元的內碼的低位元組與該第二區間之初始內碼的低位元組間的差距，並根據想要的雙位元字元的內碼的高位元組與初始內碼值的高位元組間的差距，想要的雙位元字元的內碼的內碼的低位元組與初始內碼的低位元組間的差距減去空白編碼位置數量的數值，第一區間內的編碼位置所佔據的位元總數，以及雙位元字元位元組數目加上字型資料位元組數目的總和計來算想要的雙位元字元於該儲存媒體中之實體位置；以及

若想要的雙位元字元的內碼落於該內碼系統的第三區間中，計算想要的雙位元字元的內碼的高位元組與該第三區間之初始內碼的高位元組間的差距，計算想要的雙位元字元的內碼的低位元組與該第三區間之初始內碼的低位元組間的差距，並根據想要的雙位元字元的內碼的高位元組與初始內碼值的高位元組間的差距，想要的雙位元字元的內碼的內碼的低位元組與初始內碼的低位元組間的差距減去空白編碼位置數量的數值，第一區間與第二區間內的編碼位置所佔據的位元總數，以及雙位元字元位元組數目加上字型資料位元組數目的總和計來算想要的雙位元字元於該儲存媒體中之實體位置。

6. 如申請專利範圍第 1 項所述之方法，其中該影音來源檔案係為一 MP3 (MPEG layer 3) 檔案。

7. 一種影音解碼/播放/複製系統，其係以直接螢幕顯示方法顯示雙位元字元於其一顯示器上，並藉此以更新該顯示

器的顯示畫面，該影音解碼/播放/複製系統包含：

一儲存媒體，其記錄一影音來源檔案以及至少一種雙位元字集檔案；

一直接螢幕顯示緩衝記憶體，儲存有直接螢幕顯示資料；

一記憶體；

一處理器，與該記憶體相耦接，該處理器能讀取該儲存媒體之該檔案系統，並將該檔案之名稱的內碼暫儲至該記憶體中；以及

一直接螢幕顯示單元，與該直接螢幕顯示緩衝記憶體相耦接，該直接螢幕顯示單元能經由該處理器自該雙位元字集檔案中根據該暫儲的檔案名稱的內碼擷取出所想要的雙位元字元，並配合該直接螢幕顯示資料，以直接螢幕顯示方式將所想要的雙位元字元顯示於該顯示裝置上，藉此更新該顯示裝置的顯示畫面。

8. 如申請專利範圍第 7 項所述之影音解碼/播放/複製系統，其中，該處理器更具有一檔案過濾模組，能由該儲存媒體之檔案系統中過濾出該雙位元字集檔案，以便進行雙位元字元的讀取。

9. 如申請專利範圍第 7 項所述之影音解碼/播放/複製系統，其中，該處理器更具有一字元計算模組，能利用該雙位元字集檔案所定義之一起始位置與一終止位置間，計算出各該內碼所對應之字元的位置的偏移量，藉此得到各該字元的實體位置。

10. 如申請專利範圍第 7 項所述之影音解碼/播放/複製系統，其中，該儲存媒體係為光碟片或記憶卡其中之一。
11. 如申請專利範圍第 7 項所述之影音解碼/播放/複製系統，更包含一影音來源及直接螢幕顯示字型檔偵測模組，用以偵測影音來源檔案及雙位元字集檔案的位置，並在當有數種型式的儲存媒體記錄影音來源及雙位元字集檔案可以獲得時，切換影音來源檔案及雙位元字集檔案的位置。
12. 如申請專利範圍第 7 項所述之影音解碼/播放/複製系統，其中，該影音來源檔案係為一 MP3 檔案。

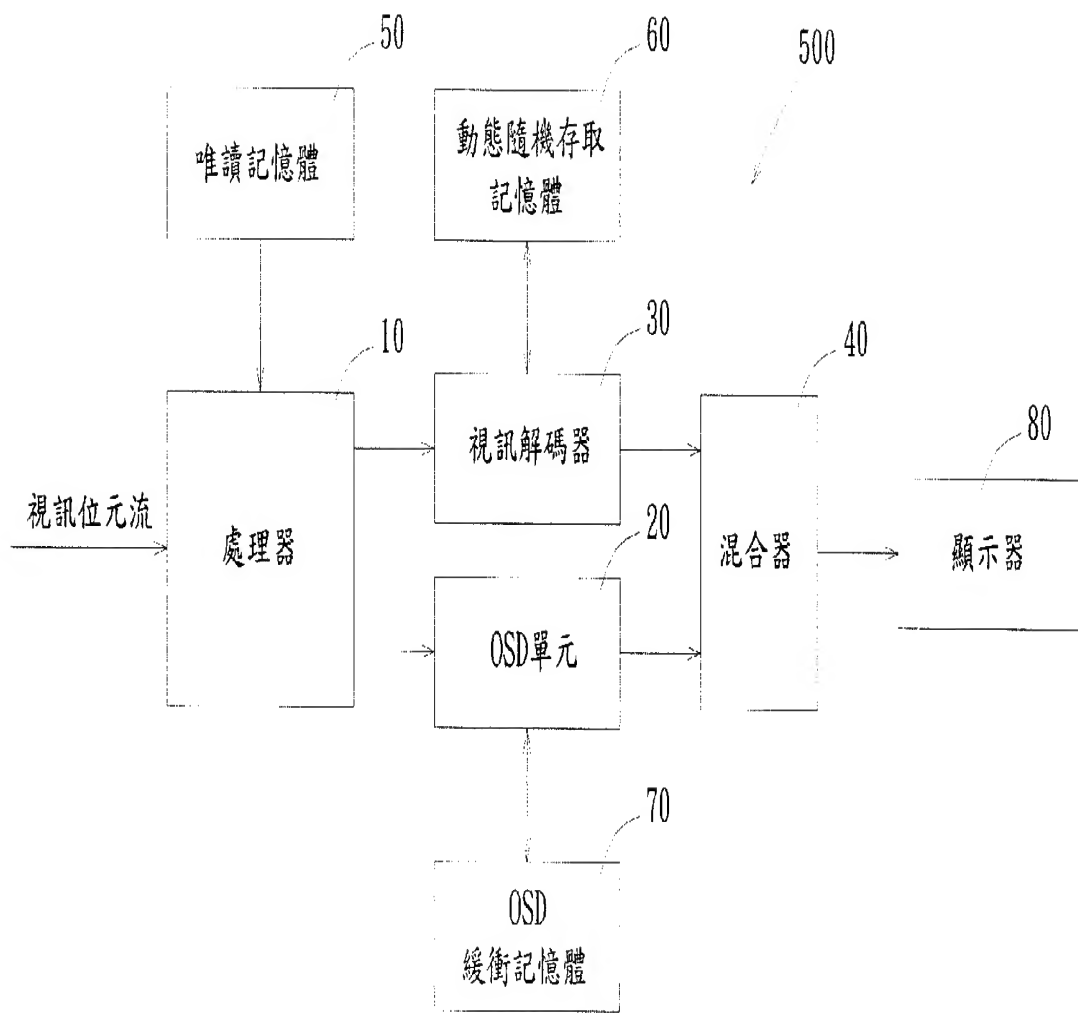


圖 1

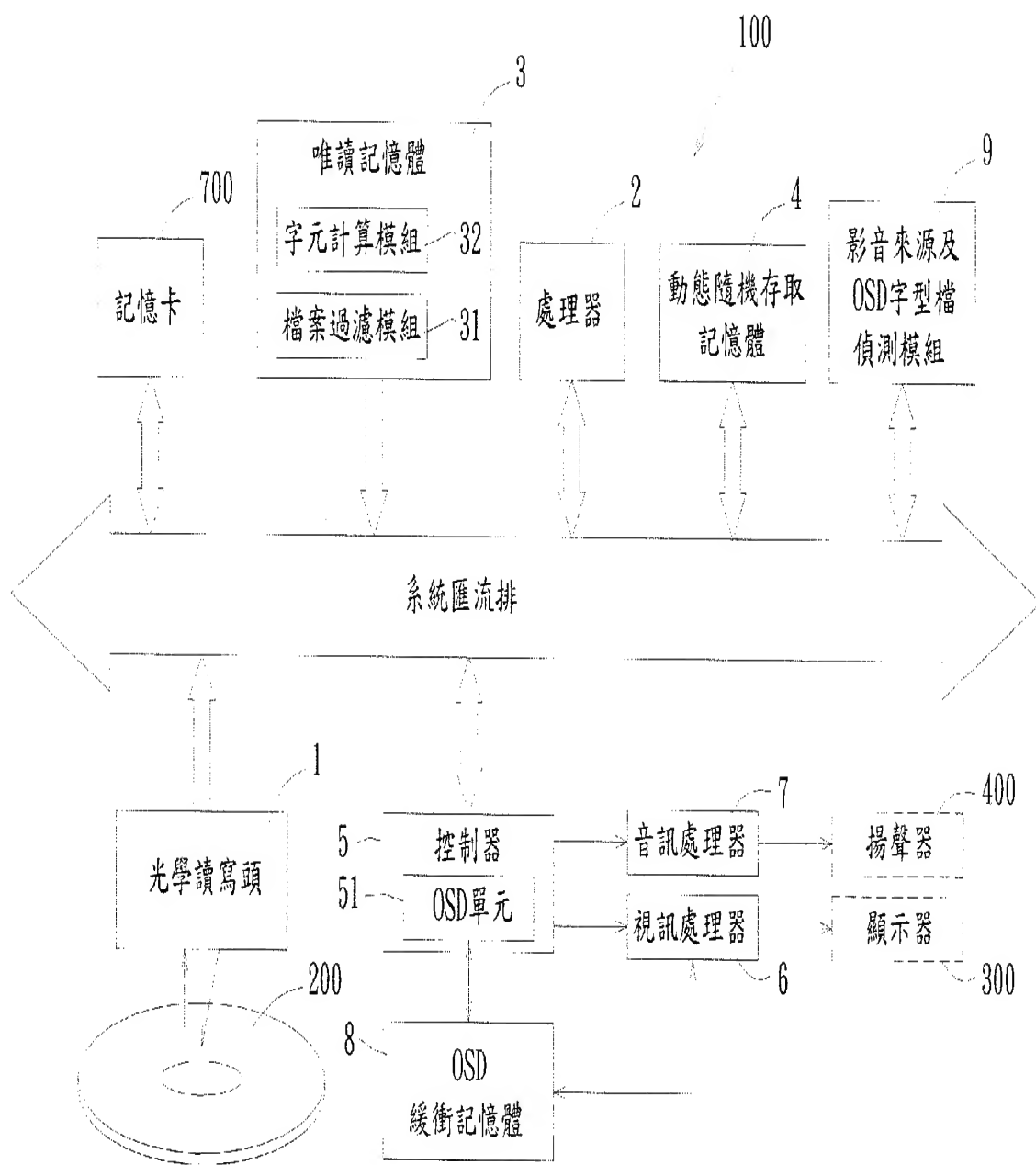


圖 2

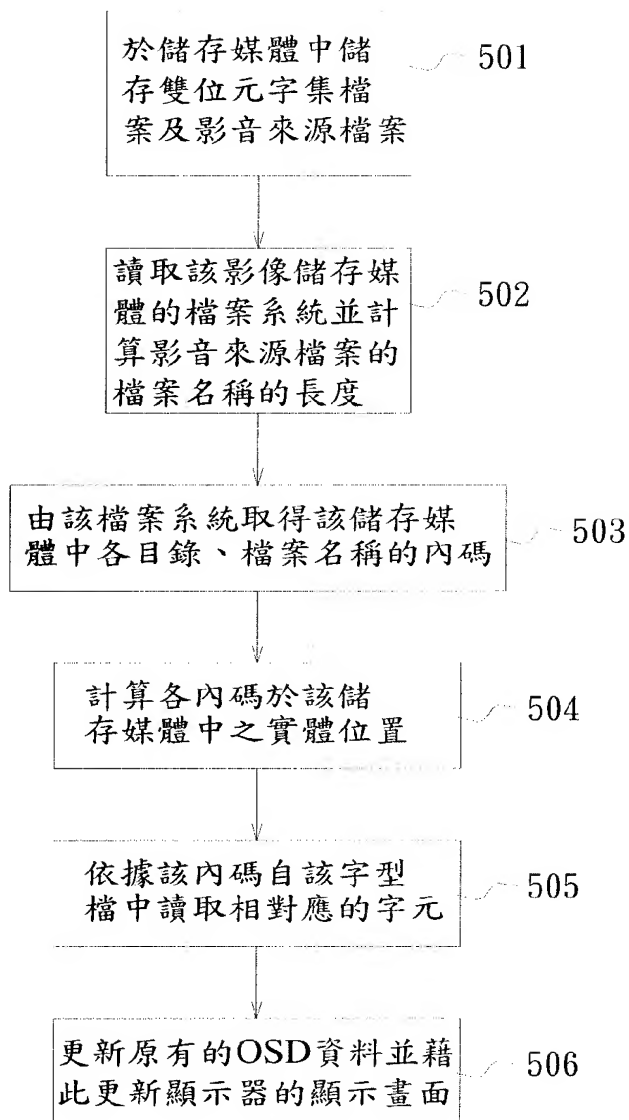


圖 5